ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)
Vol. 2, Issue 2 (Apr. - June 2015)
International Journal of Advanced Research in Education Technology (IJARET)

Preserving User Privacy in Personalized Web Search Service

[I]M.Dhivya, [II]S.Nagaraj

Abstract

The user profiles, descriptions of user interests, can be used by search engines to provide personalized search results. Many approaches to creating user profiles capture user information through proxy servers or desktop bots. These both require participation of the user to install the proxy server or the bot. In this study, we explore the use of a less-invasive means of gathering user information for personalized search. In particular, we build user profiles based on activity at the search site itself andstudy the use of these profiles to provide personalized search results. In our study, we implemented a wrapper for Google to examine different sources of information on which to base the user profiles: queries and snippets of examined search results. These user profiles were created by classifying the information into concepts from the Open Directory Project concept hierarchy and then used to re-rank the search results. User feedback was collected to compare Google's original rank with our new rank for the results examined by users. They found that queries were as effective as snippets when used to create user profiles and that our personalized re-ranking resulted in a improvement in the rank order of the user-selected results.

1. Introduction

Personalization has been a very active research field in the last several years and user profile construction is an important component of any personalization system. Explicit customization has been widely used to personalize the look and content of many web sites, personalized search approaches focus on implicitly building and exploiting user profiles. Companies that provide marketing data report that search engines are utilized more and more as referrals to web sites, compared to direct navigation and web links. As search engines perform a larger role in commercial applications, the desire to increase their effectiveness grows. However, search engines are affected by problems such as ambiguity and results ordered by web site popularity rather than user interests.

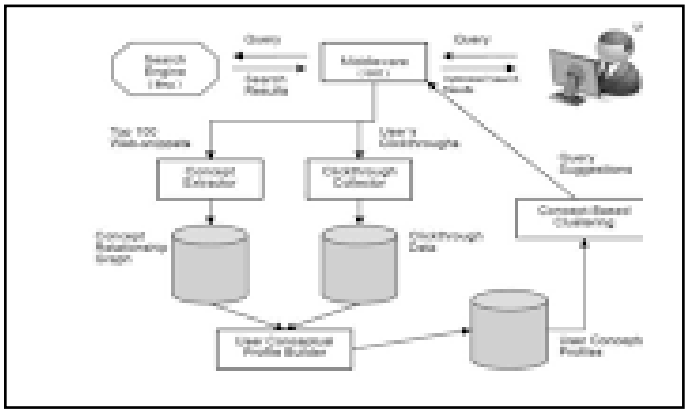
They based on building user profiles based on the user's interactions with a particular search engine. For this purpose, Implemented GoogleWrapper: a wrapper around the oogle search engine, that logs the queries, search results, and clicks on a per user basis. This information was then used to create user profiles and these profiles were used in a controlled study to determine their effectiveness for providing personalized search results.

They conducted in three phases:

1. Collecting information from users. All searches, for which at least one of the results was clicked were logged per user.
2. Creation of user profiles. Two different sources of information were identified for this purpose: all queries submitted for which at least one of the results was visited and all snippets visited. Two profiles were created out of either queries and snippets.
3. Evaluation: the profiles created were used to calculate a new rank of results browsed by users. The average of this rank was compared with Google's rank.

II. Existing System

The solutions to PWS can generally be categorized into two types, namely click-log-based methods and profile-based ones. The click-log based methods are straightforward they simply impose bias to clicked pages in the user's query history. Although this strategy has been demonstrated to perform consistently and considerably well , it can only work on repeated queries from the same user, which is a strong limitation confining its applicability. In contrast, profile-based methods improve the search experience with complicated user-interest models generated from user profiling techniques. Profile-based methods can be poten-tially effective for almost all sorts of queries, but arereported to be unstable under some circumstances . The existing profile-based PWS do not support runtime profiling.

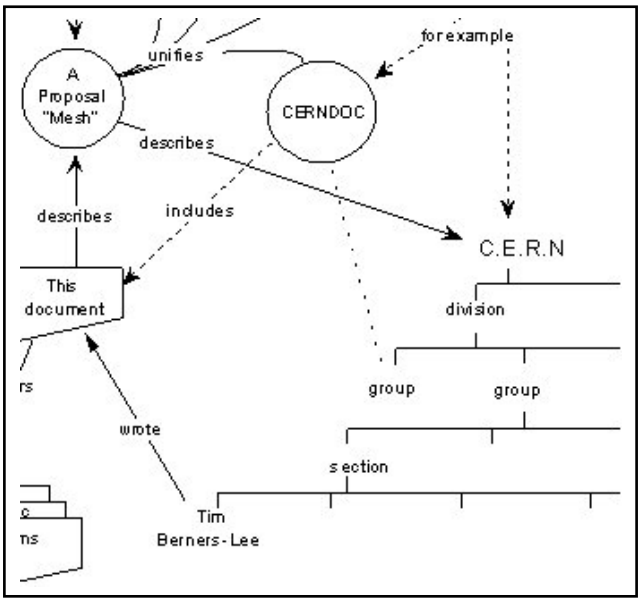


The existing methods do not take into account the customization of privacy requirements.A user profile is typically generalized for only once offline, and used to personalize all queries from a same user indiscriminatingly. Such "one profile fits all strategy certainly has drawbacks given the variety of queries. One evidence reported in is that profile-based personalization may not even help to improve the search quality for some ad hoc queries, though exposing user profile to a server has put the user's privacy at risk.

III. Proposed System

A privacy-preserving personalized web search framework UPS, which can generalize profiles for each query according to user-specified privacy requirements.Relying on the definition of two conflicting metrics, namely personalization utility and privacy risk, for hierarchical user profile, we formulate the problem of privacy-preserving personalized search as #-Risk Profile Generalization, with its N P-hardness proved.

ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)



They develop two simple but effective generalization algorithms, GreedyDP and GreedyIL, to support runtime profiling. While the former tries to maximize the discriminating power (DP), the latter attempts to minimize the information loss (IL). By exploiting a number of heuristics, GreedyIL out performs GreedyDP significantly.They provide an inexpensive mechanism for the client to decide whether to personalize a query.

Increasing usage of personal and behaviour information to profile its users, which is usually gathered implicitly from query history, browsing history, click-through data bookmarks, user documents, and so forth.The framework allowed users to specify customized privacy requirements via the hierarchical profiles.

In addition, UPS also performed online generalization on user profiles to protect the personal privacy without compromising the search quality.

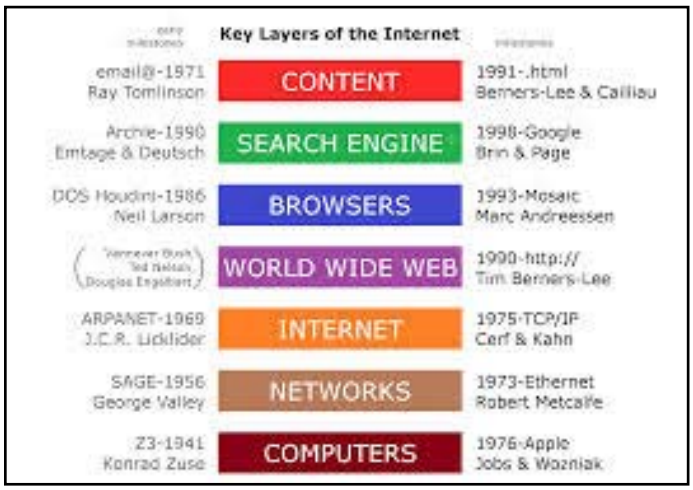
IV. Background

A. Ontologies and Semantic Web

According to Gruber, an ontology is a "specification of a conceptualization". Ontologies can be defined in different ways but they all represent a taxonomy of concepts along with the relations between them. In the context of the World Wide Web, ontologies are important because they formally define terms shared between any type of agents without ambiguity, allowing information to be processed automatically and accurately.

OntoSeek is an example of system based on ontologies. Utilizing information sources such as product catalogs and yellow pages it applies conceptual graphs to represent both queries and resources.

The expression "Semantic Web" was introduced by ETAI (Electronic Transactions on Artificial Intelligence) in 2000 to describe the extension of the Web to deal with the meaning of available content rather than just its syntactic form.



Many XMLbased projects such as Resource Descriptor Framework (RDF), Notation 3 (N3), and OWL started from there and each aims to define a syntax capable of describing and/or manipulating ontologies. One of the main bottlenecks in the evolution of the Web along these lines is the amount of manual effort usually required to create, maintain, and use ontologies. Our approach shares many of the same goals as the Semantic Web, however we focus on automatic techniques wherever possible.

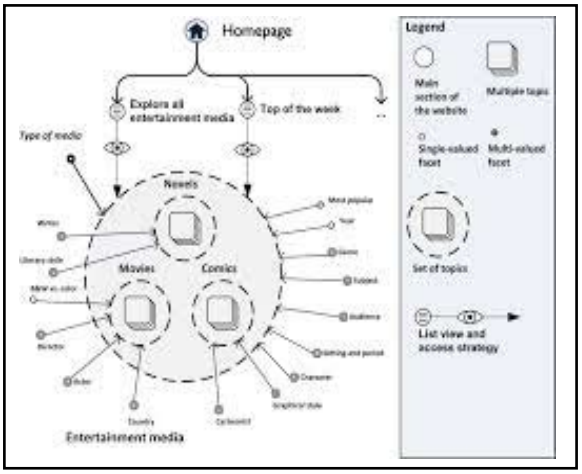
B. Personalization

Personalization is the process of presenting the right information to the right user at the right moment. In order to learn about a user, systems must collect information about them, analyze the information, and store the results of the analysis in a user profile. Information can be collected from users in two ways: explicitly, for example asking for feedback such as preferences or ratings; and implicitly, for example observing user behaviors such as the time spent reading an online document. Explicit construction of user profiles has several drawbacks. The user provide inconsistent or incorrect information, the profile built is static whereas the user's interests may change over time, and the construction of the profile places a burden on the user that they may not wish to accept. Thus, many research efforts are underway to implicitly create accurate user profiles .

User browsing histories are the most frequently used source of information about user interests. Trajkova and Gauch use this information to create user profiles represented as weighted concept hierarchies. The user profiles are created by classifying the collected Web pages with respect to a reference ontology.

V. Approach

Our study investigates the effectiveness of personalized search based upon user profiles constructed from user search histories. GoogleWrapper is used to monitor users activities on the search site itself in order to gather individual user information such as queries submitted, results returned (titles and snippets), and Web pages selected from results retrieved. This per-user information is classified into a concept hierarchy based upon the Open Directory Project producing conceptual user profiles.

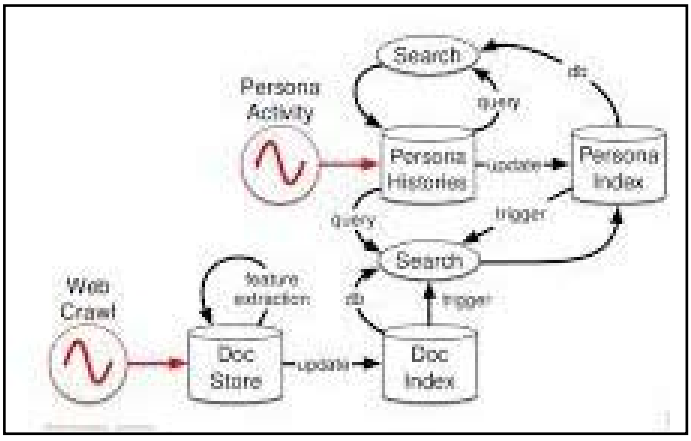


Search results are also classified into the same concept hierarchy, and the match between the user profile concepts and result concepts are used to re-rank search results. It believe this approach has several advantages. User interests are collected in a completely non-invasiveway,search personalization is based upon data readily available to the search engine, and the system effectiveness can be evaluated by monitoring user activities rather than requiring explicit judgments or feedback.

A. System Architecture

GoogleWrapper: a wrapper for Google that implicitly collects information from users. Google APIs and nusoap library were used for the implementation.

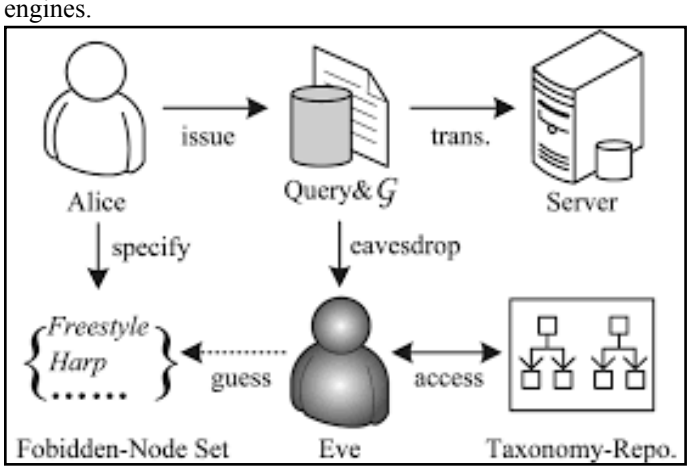
Users register with their email addresses in order to create a cookie storing their userID on their local machines. If the cookie was lost, GoogleWrapper notified the user and they could login to reset the cookie. When queries are submitted by users, GoogleWrapper logs the query and the userID and then forwards the query to the Google search engine. It intercepts the search engine results, logs them, re-ranks them, and then displays them to the user.



The classifier from KeyConcept , a conceptual search engine, is used to classify queries, snippets for each user as well as the search engine results. This vector space model classifier implements a k nearest neighbors algorithm..

B. User Profiles

User profiles are represented as a weighted concept hierarchy. The concepts hierarchy is created from 1,869 concepts in the top 3 levels of the Open Directory Project and the weights represent the amount of user interest in the concept. The concept weights are assigned by classifying textual content collected from the user into the appropriate concepts using a vector space classifier and the k- nearest neighbor algorithm. The weights assigned by the classifier are accumulated over the text submitted. They constructed user profiles from Web pages browsed by the user, however, this study focused on using the user's search history rather than their browsing history, information more easily available to search engines.



They evaluate the effectiveness of profiles built from user queries with those built from snippets of userselected results. Each query or snippet was classified, resulting in a list of concepts and weights in decreasing order of weight. Since the number of concepts per item to add to the profile was unknown, preliminary analysis of the classifier results for queries submitted by 8 different users. By manually judging the classifier results as relevant or not, we determined that the concepts assigned per query were relevant 75% of the time, dropping dramatically after that. A similar analysis for snippets determined that the top 5 classifier results were reasonably accurate.

C. Personalized Search

A user submits a query to the search engine, and the titles, summaries and ranks results are obtained. The results are re-ranked using a combination of their original rank and their conceptual similarity to the user's profile. The search result titles and summaries are classified to create a document profile in the same format as the user profile. The document profile is then compared to the user profile to calculate the conceptual similarity between each document and the user's interests.

The documents are re-ranked by their conceptual similarity to produce their conceptual rank. The final rank of the document is calculated by combining the conceptual rank with Google's original rank using the following weighting scheme:

FinalRank = α * ConceptualRank + (1-α) * GoogleRank

α has a value between 0 and 1. When α has a value of 0, conceptual rank is not given any weight, and it is equivalent to the original rank assigned by Google. If α has a value of 1, the search engine ranking is ignored and pure conceptual rank is considered. The conceptual and search engine based rankings can be blended in different proportions by varying the value of α.

D. The GreedyDP Algorithm

Given the complexity of our problem, a more practical solution would be a near-optimal greedy algorithm. As preliminary, we introduce an operator called prune-leaf, which indicates the removal of a leaf topic t from a profile. Àt Formally, we denote by Gi _t Giþ1 the process of pruning leaf t from Gi to obtain Gi+1 . Obviously, the optimal profile G can be generated with a finite-length transitive closure of prune-leaf.

The first greedy algorithm GreedyDP works in a bottom- up manner. Starting from G0 , in every ith iteration, GreedyDP chooses a leaf topic t for pruning, trying to maximize the utility of the output of the current iteration, namely Gi+1 . During the iterations, we also maintain a best- profile-so-far, which indicates the Gi+1 having the highest discriminating power while satisfying the α-risk constraint.

The iterative process terminates when the profile is generalized to a root-topic. The best-profile-so-far will be the final result of the algorithm. The main problem of GreedyDP is that it requires recomputation of all candidate profiles generated from attempts of prune-leaf on all t.This causes significant memory requirements and computational cost.

E. The GreedyIL Algorithm

The GreedyIL algorithm improves the efficiency of the generalization using heuristics based on several findings. One important finding is that any prune-leaf operation reduces the discriminating power of the profile. In other words, the DP displays monotonicity by prune-leaf.

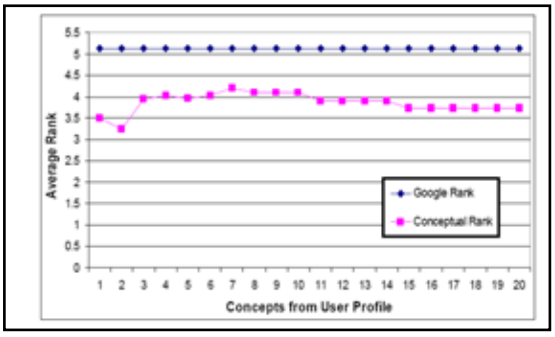
VI. Validation of Experiental

Monitored the search activities of six volunteers for approximately six months. All queries submitted per user were divided into 40 training queries, those used to create profiles, and 5 testing queries, those used to evaluate the profiles. Five testing queries were selected and up to 6 profiles were evaluated by their effectiveness for personalizing the search results measured by comparing the rank order of the user-selected results with and without re-ranking based on the profile.

A preliminary study and examined randomly selected queries from different users.The user-selected results occurred in the first Google results and no result after the tenth result was ever selected. The number of clicks on the first result was also much higher than the number of clicks on the second one. The samedecrease was observed between the second and the third results. The user judgments were affected by Google's rank so, for this study, randomized the search engine results before presentation to the user. Since all results selected occurred within the first page, only randomized the first ten results.

The user clicks thus collected were analyzed later to compare how Google ranked the selected result versus how our system would have ranked it based upon the user profile.

The first variable we investigated was the number of training queries necessary to create a profile based upon the query text alone. As mentioned in returned by the classifier for each query. We created user profiles using training sets of queries. A second variable studied was the number of concepts from the resulting profile to use when calculating the similarity between the profile and the document.



VII. Implementation Issues

A. Inverted-Index of Topics

Many of the publicly available repositories allow for manual tagging and editing on each topic . These textual data associated with the topics comprise a document repository D(R), so that each leaf topic t∈R finds its associated document set D(t)< D(R), which describes t itself. For simplicity, They assume that d(t1)∩d(t2). In other words, each document in D(R) is assigned to only one leaf topic. Thus, for each leaf topic t 2 R, it is possible to generate an inverted-index, denoted by $\mathcal{I}$(t), containing entries like for all documents in D(t).

B. Topic Detection in R

During Offline-1 procedure, we need to detect the respective topic in R for each document d € D. A naive method is o compute for each pair of d and t € R their relevance with discriminative naive Bayesian classifier

C. Query-Topic Relevance

The computation of query-topic relevanc during online-1is straightforward. Given a query q, we retrieve from inverted index I[Top] the documents relevant to q using the conventional approach. These documents are then grouped by their respective topics. The relevance of each topic is then computed as the number of documents contained in each topic. We note that the relevance metric used in our implementation is very simple and fast to evaluate. It can easily be replaced by more complicated versions

VIII. Conclusion and Future Work

The user profiles based on implicitly collected information, specifically the queries submitted and snippets of user-selected results. They were able to demonstrate that information readily available to search engines is sufficient to provide significantly improved personalized rankings. They found that using a profile built from the queries produced an improvement of the rank of the selected result. A user profile built from snippets user-selected results showed a larger, but not significant.

The snippetbased profile also improved more queries and hurt fewer so there is some indication that it is a slightly more accurate profile. Our best results occurred when conceptual ranking considered only one concept from the query-based profile, and two from the snippet-based profile. This may be because the training and testing queries came from a relatively short window of time and users were working in a focused manner. However, the ranking improvements hold fairly steady across the evaluated range of 1 – 20 concepts used. For personalized results over a broader range of user queries, it would be safer to use more concepts from the profile.

References

[1] *D. Billsus, M.J. Pazzani. A hybrid user model for news story classification. In proceedings of the seventh international conference on User modeling, Banff, Canada, pp. 99 – 108, 1999.*

[2] *J. Chaffee, S. Gauch. Personal Ontologies for Web Navigation. In Proceedings of the 9th International Conference on Information and Knowledge Management. (CIKM), pp 227 – 234, 2000.*

[3] *V. Challam. Contextual information retrieval using ontology*

based user profile. Master's thesis, University of Kansas, Lawrence, KS, 2004.

[4] P.K. Chan. A Non-Invasive Learning Approach to Building Web User Profiles. KDD-99 Workshop on web usage analysis and user profiling, pp. 7 – 12, 1999.

[5] M. Chau, D. Zeng, H. Chen. Personalized spiders for web search and analysis. In Proceedings of the 1st ACM-IEEE Joint Conference on Digital Libraries, pp 79 - 87, 2001.

[6] C.C. Chen, M.C. Chen, Y. Sun. PVA: a self-adaptive personal view agent. In proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining 2001, San Francisco, California, pp. 257 – 262, 2001.